VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

Fall Semester 2023-2024

Continuous Assessment Test – 1

SLOT: B1+TB1, B2+TB2

Programme Name &Branch: B.Tech & BCB, BCE, BCI, BCT, BDS, BKT

Course Name & code: BCSE303L & Operating Systems

Question Paper & Answer Key

Q.No.	Question	Max Marks	CO	BL
1.	Describe the various methods used for operating system Structure. Also discuss the advantages and disadvantages of each structuring method.	10	CO1	BL2
2.	a) What is a system call? Why are system calls necessary? Illustrate the methods to pass the parameters of system calls to the OS.	5	CO1	BL2
	b) Categorize the following instructions into privileged instructions and non-privileged instructions. Also mention whether the instruction to be executed under user mode or kernel mode. i) Reading system time ii) Clear Memory iii) Opening and reading a file iv) Set the timer v) Performing arithmetic operation	5	CO1	BL2
3.	What are the differences between user-level threads and kernel-level threads? Mention the advantages of using multithreaded programming in multicore architecture systems. Also illustrate the different multi-threading models.	10	CO1	BL2

<table>
<tr><td>4.</td><td>Consider a system with five processes (P1, P2, P3, P4, P5), all arriving at time zero, with total execution time (includes CPU Burst time and I/O Burst time) of 25, 15, 20, 10 and 5 milliseconds respectively. Each process spends 20% of execution time doing I/O and 80% of time doing computation. The operating system uses SJF and FCFS scheduling algorithms to schedule the processes by considering only CPU burst time of each process. Calculate average turnaround time and average waiting time for both the algorithms.</td><td>10</td><td>CO2</td><td>BL3</td></tr>
<tr><td>5.</td><td>Assume the following workload in a system:
<table><tr><th>Process</th><th>Arrival Time (ms)</th><th>Burst Time (ms)</th><th>Priority</th></tr><tr><td>P1</td><td>5</td><td>5</td><td>0</td></tr><tr><td>P2</td><td>4</td><td>2</td><td>7</td></tr><tr><td>P3</td><td>3</td><td>7</td><td>5</td></tr><tr><td>P4</td><td>0</td><td>4</td><td>10</td></tr><tr><td>P5</td><td>3</td><td>5</td><td>5</td></tr></table>Draw the Gantt chart illustrating the execution of these processes using Round robin scheduling algorithm (Time Quantum= 3 ms) and priority scheduling algorithm, also Calculate the average waiting time and average turnaround time.</td><td>10</td><td>CO2</td><td>BL3</td></tr>
</table>

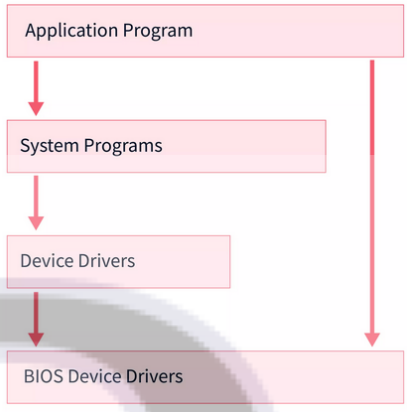
Answer Key

1. OS Structuring Methods: (10 Marks)

- Simple Structure
- Monolithic Structure
- Layered Approach Structure
- Micro-kernel Structure

Simple Structure (2.5 marks)

- It is the simplest Operating System Structure and is not well defined;
- It can only be used for small and limited systems.
- In this structure, the interfaces and levels of functionality are well separated; hence programs can access I/O routines which can cause unauthorized access to I/O routines.



Advantages of Simple Structure

- It is easy to develop because of the limited number of interfaces and layers.
- Offers good performance due to lesser layers between hardware and applications.

Disadvantages of Simple Structure

- If one user program fails, the entire operating system crashes.
- Abstraction or data hiding is not present as layers are connected and communicate with each other.
- Layers can access the processes going in the Operating System, which can lead to data modification and can cause Operating System to crash.

Monolithic Approach (2.5 marks)

- Functionality of the OS is invoked with simple function calls within the kernel, which is one large program.
- Device drivers are loaded into the running kernel and become part of the kernel.

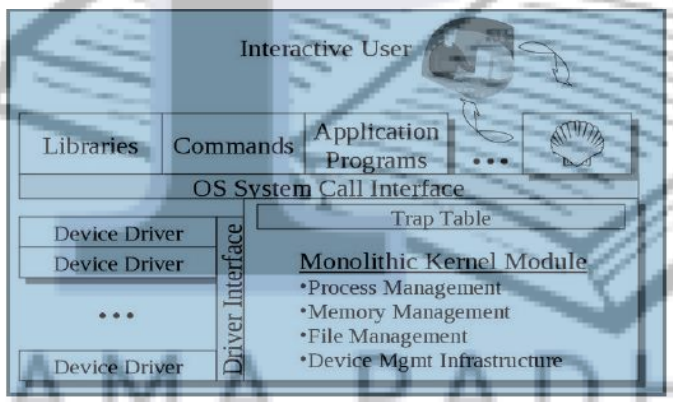


Figure : A monolithic kernel, such as Linux and other Unix systems

Advantages of Monolithic structure:

- It is simple to design and implement because all operations are managed by kernel only, and layering is not needed.
- As services such as memory management, file management, process scheduling, etc., are implemented in the same address space, the execution of the monolithic kernel is relatively fast as compared to normal systems. Using the same address saves time for address allocation for new processes and makes it faster.

Disadvantages of Monolithic structure:

- If any service in the monolithic kernel fails, the entire System fails because, in address space, the services are connected to each other and affect each other.
- It is not flexible, and to introduce a new service

Layered Approach (2.5 marks)

- This approach breaks up the operating system into different layers.
- This allows implementers to change the inner workings, and increases modularity.
- As long as the external interface of the routines don't change, developers have more freedom to change the inner workings of the routines.
- With the layered approach, the bottom layer is the hardware, while the highest layer is the user interface.
- **Advantages:**
  - The main advantage is simplicity of construction and debugging.
- **Disadvantages:**
  - The main difficulty is defining the various layers.
  - The main disadvantage is that the OS tends to be less efficient than other implementations.

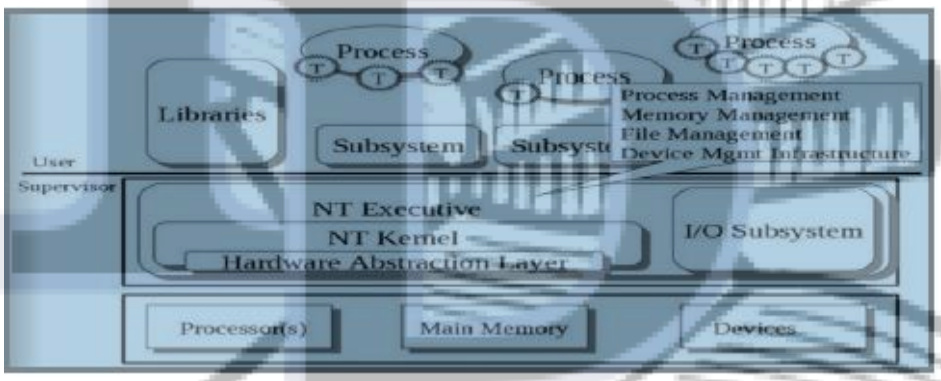


Figure: Layered Model

Microkernels (2.5 marks)

- This structures the operating system by removing all nonessential portions of the kernel and implementing them as system and user level programs.
- Generally, they provide minimal process and memory management, and a communications facility.
- Communication between components of the OS is provided by message passing.
- **Advantages:**
  ✓ Extending the operating system becomes much easier.
  ✓ Any changes to the kernel tend to be fewer, since the kernel is smaller.
  ✓ The microkernel also provides more security and reliability.
- **Disadvantages:**
  ✓ Main disadvantage is poor performance due to increased system overhead from message passing

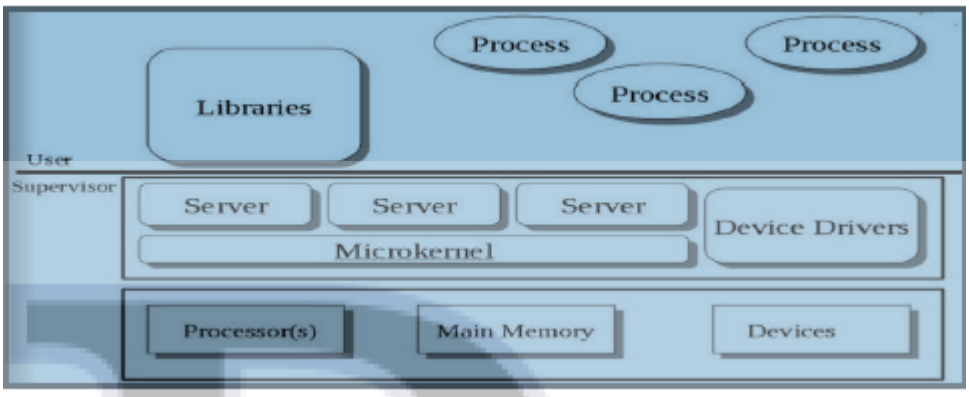


Figure: A Microkernel architecture

2. (a) System Calls: (5 Marks)

Definition: (1 Mark)

- A system call is a method for a computer program to request a service from the kernel of the operating system on which it is running.
- A system call is a method of interacting with the operating system via programs. A system call is a request from computer software to an operating system's kernel.

Need: (1 Mark)

- System calls allow user-level processes to request services of the operating system
- Simplicity – No need to write complex code

Parameter passing methods for System Call: (3 Marks)

1) Parameters should be pushed on or popped off the stack by the operating system.
2) Parameters can be passed in registers.
3) When there are more parameters than registers, it should be stored in a block, and the block address should be passed as a parameter to a register.

2. (b) Instruction's Category: (5 Marks)

S.No	Instruction	Privileged or Non-Privileged Instruction	Mode (Kernel mode or User Mode)
i)	Reading system time	Non-Privileged	User Mode
ii)	Clear Memory	Privileged	Kernel Mode
iii)	Opening and reading a file	Privileged	Kernel Mode
iv)	Set the timer	Privileged	Kernel Mode
v)	Performing arithmetic operation	Non-Privileged	User Mode

3. Threads: (10 Marks)

Difference between User Level and Kernel Level Thread: (2 marks)

User Level Thread	Kernel Level Thread
User-level threads are faster to create and manage.	Kernel level threads are slower to create and manage.
Implemented by a thread library user level.	Operating system support directly to Kernel threads.
User level thread can run on any operating system.	Kernel level threads are specific to .the operating system.
Support provided at the user level called user-level thread	Support may be provided by Kernel is called Kernel level threads.
Multithread applications cannot take advantage of multiprocessing.	Kernel routines themselves can be multithreaded.
Implementation of User threads is easy.	Implementation of Kernel thread is complicated.
Context switch time is less.	Context switch time is more.
Context switch requires no hardware support.	Hardware support is needed.
Example: User-thread libraries include POSIX Pthreads, Mach C-threads, and Solaris 2 UI-threads.	Example: Windows NT, Windows 2000, Solaris 2, BeOS, and Tru64 UNIX (formerly Digital UNIX)-support kernel threads.

Advantages: (2 marks)

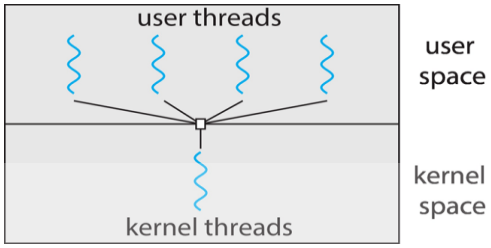
- Responsiveness
- Resource sharing
- Economy
- Improved throughput
- Utilization of Multiprocessor/Multi-core Architecture – If a system has only one process which is also single threaded process, then there is no effective utilization of all the processors. If the process is multi-threaded, all the processors can be effectively utilized.

Multithreading Models (6 marks)

- **Many-to-One**
- **One-to-One**
- **Many-to-Many**

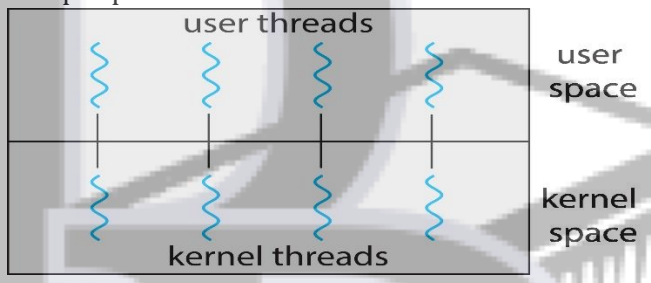
Many-to-One:

- Many user-level threads mapped to single kernel thread
- One thread blocking causes all to block
- Multiple threads may not run in parallel on multicore system because only one may be in kernel at a time
- Few systems currently use this model



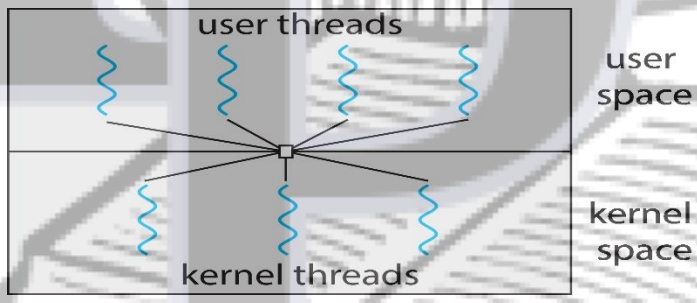
One-to-One:

- Each user-level thread maps to kernel thread
- Creating a user-level thread creates a kernel thread
- More concurrency than many-to-one
- Number of threads per process sometimes restricted due to overhead



Many-to-Many Model:

- Allows many user level threads to be mapped to many kernel threads
- Allows the operating system to create a sufficient number of kernel threads



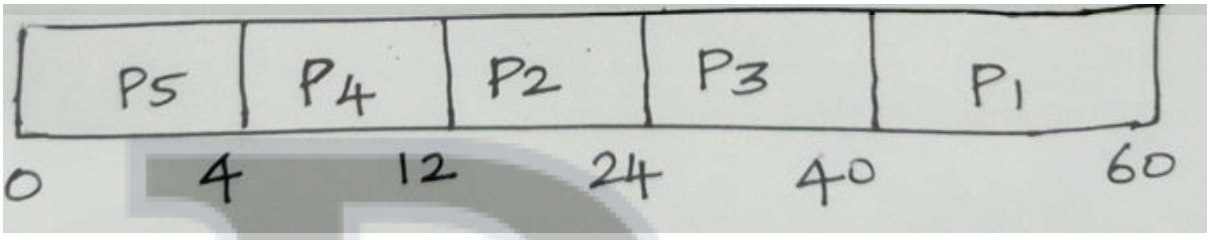
4. Solution: (10 marks)

Calculation of CPU Burst Time: (2 Marks)

Process	Total Execution Time (ms)	CPU Burst Time [80% of total execution time] (ms)
P1	25	20
P2	15	12
P3	20	16
P4	10	8
P5	5	4

SJF: (4 Marks)

Gantt Chart:



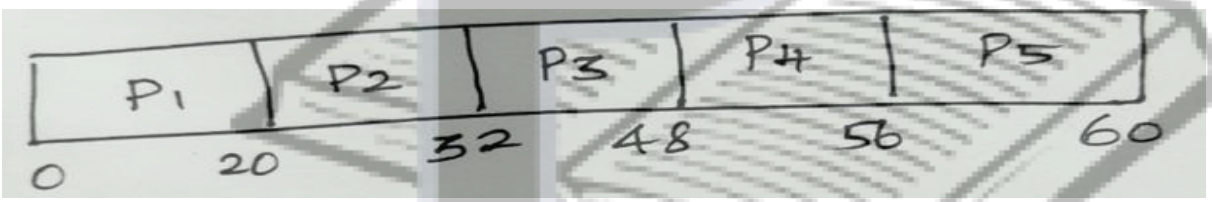
Process	Waiting Time	Turnaround Time
P1	40	60
P2	12	24
P3	24	40
P4	4	12
P5	0	4

Average waiting time: 16 ms

Average Turnaround time: 28 ms

FCFS: (4 Marks)

Gantt Chart:



Process	Waiting Time	Turnaround Time
P1	0	20
P2	20	32
P3	32	48
P4	48	56
P5	56	60

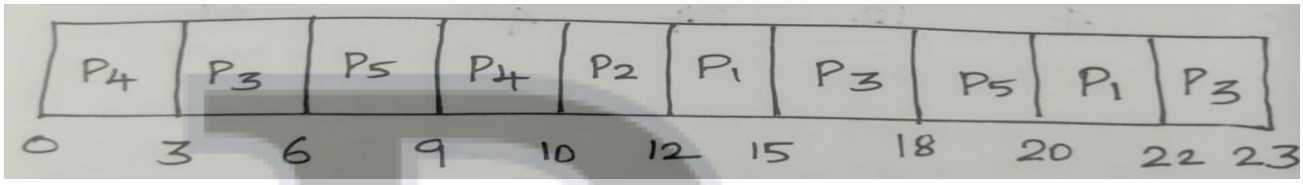
Average waiting time: 31.2 ms

Average Turnaround time: 43.2 ms

5. Solution: (10 Marks)

Round Robin (Time Slice=3 ms): (4 Marks)

Gantt Chart:



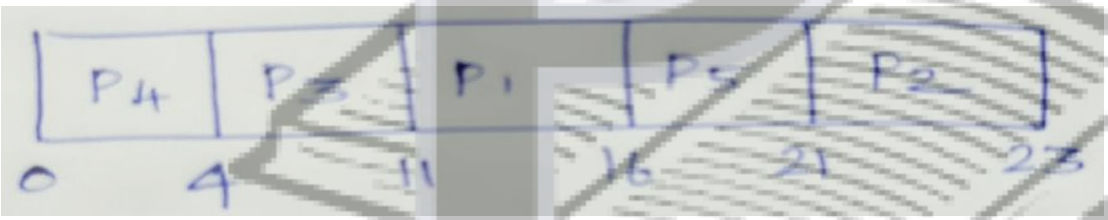
Process	Waiting Time	Turnaround Time
P1	12	17
P2	6	8
P3	13	20
P4	6	10
P5	12	17

Average waiting time: 9.8 ms

Average Turnaround time: 14.4 ms

Priority (Non preemptive) : (3 Marks)

Gantt Chart:



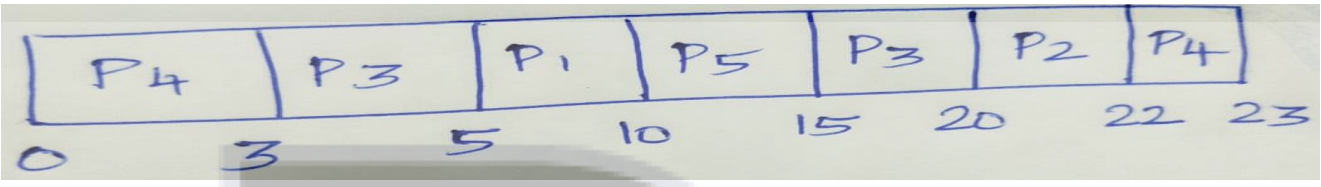
Process	Waiting Time	Turnaround Time
P1	6	11
P2	17	19
P3	1	8
P4	0	4
P5	13	18

Average waiting time: 7.4 ms

Average Turnaround time: 12 ms

Priority (Preemptive): (3 Marks)

Gantt Chart:



Process	Waiting Time	Turnaround Time
P1	0	5
P2	16	18
P3	10	17
P4	19	23
P5	7	12

Average waiting time: 10.4 ms

Average Turnaround time: 15 ms